

MULTIPASS AUTOMATA AND GROUP WORD PROBLEMS

TULLIO CECCHERINI-SILBERSTEIN, MICHEL COORNAERT, FRANCESCA FIORENZI,
PAUL E. SCHUPP, AND NICHOLAS W. M. TOUIKAN

ABSTRACT. We introduce the notion of multipass automata as a generalization of pushdown automata and study the classes of languages accepted by such machines. The class of languages accepted by deterministic multipass automata is exactly the Boolean closure of the class of deterministic context-free languages while the class of languages accepted by nondeterministic multipass automata is exactly the class of poly-context-free languages, that is, languages which are the intersection of finitely many context-free languages. We illustrate the use of these automata by studying groups whose word problems are in the above classes.

1. INTRODUCTION

The main purpose of this paper is to introduce a very natural machine model, that of *multipass automata*. These are essentially like pushdown automata except that they are able to read the input tape several times. It turns out that the class \mathcal{DM} of languages accepted by deterministic multipass automata is exactly the Boolean closure of the class of deterministic context-free languages, which we denote by \mathcal{BDC} .

The class \mathcal{NM} of languages accepted by nondeterministic multipass automata is exactly the class \mathcal{PCF} of *poly-context-free* languages, that is, languages which are the intersection of finitely many context-free languages. It follows from [20] that the following strict inclusions hold

$$(1.1) \quad \mathcal{DM} = \mathcal{BDC} \subsetneq \mathcal{NM} = \mathcal{PCF} \subsetneq \mathcal{BC}$$

where \mathcal{BC} denotes the Boolean closure of all the context-free languages. Wotschke also proved that the classes of context-free languages and deterministic multipass languages are incomparable in the sense that neither one is contained in the other. We also mention that Wotschke [21] introduced a notion of *Boolean acceptance* and presented a characterization of \mathcal{BC} and of \mathcal{PCF} in terms of Boolean acceptance.

Our basic references about context-free languages are the monographs by Harrison [10] and by Hopcroft and Ullman [12].

As a motivating example for multipass automata, consider the word problem for the free abelian group of rank two with presentation $G = \langle a, b; ab = ba \rangle$. The associated word problem is the language consisting of all words over the alphabet $\Sigma = \{a, a^{-1}, b, b^{-1}\}$ which

Date: July 7, 2015.

2000 Mathematics Subject Classification. 03B25, 05C05, 37B10, 37B15, 68Q70, 68Q80.

Key words and phrases. pushdown automaton, multipass automaton, context-free language, finitely generated group, the Word Problem.

have exponent sum 0 on both a and b . The pumping lemma for context-free languages shows that this word problem is not context-free. However, this word problem is accepted by a multipass automaton M working as follows. Given an input word $w \in \Sigma^*$, on the first pass M checks if the exponent sum on a in w is 0 and, on the second pass, M checks if the exponent sum on b in w is 0. Then M accepts w at the end of the second pass if and only if both conditions are met.

In this paper we focus on applying multipass automata to study group word problems, although we believe that they will be useful in many other areas. Brough [2] studied the class \mathcal{PG} of finitely generated groups whose word problem is a poly-context-free language. Holt, Rees, Röver and Thomas [11] studied the class \mathcal{CG} of finitely generated groups whose word problem is the complement of a context-free language. This class has several properties in common with the class of poly-context-free groups. But to show that groups are in \mathcal{CG} one generally complements nondeterministic context-free languages, so there are groups in \mathcal{CG} which are not in \mathcal{PG} , for example the standard restricted wreath product $\mathbb{Z} \wr \mathbb{Z}$ [2].

The Muller-Schupp theorem [15] shows that a finitely generated group G has context-free word problem if and only if it is virtually free, that is, G contains a free subgroup of finite index. Since the word problem for a finitely generated virtually free group is a deterministic context-free language, nondeterminism does not increase the class of finitely generated groups with context-free word problem. This raises the question of the situation for \mathcal{PG} .

Brough [2] conjectured that the class \mathcal{PG} coincides with the class \mathcal{D} of (finitely generated) groups which are virtually a finitely generated subgroup of a direct product of free groups. We define \mathcal{BDG} to be the class of finitely generated groups whose word problem is in \mathcal{BDC} , the Boolean closure of deterministic context-free languages. All groups in \mathcal{D} are in \mathcal{BDG} so the conjecture would again show that nondeterminism does not increase the class of groups considered.

We will prove that if $G \in \mathcal{BDG}$ (respectively \mathcal{PG}) and S is a subgroup of finite index, and φ is an automorphism of G of finite order with $\varphi(S) = S$ then the HNN-extension

$$H = \langle G, t; tst^{-1} = \varphi(s), s \in S \rangle$$

is again in \mathcal{BDG} (respectively \mathcal{PG}). This theorem seems to us to use the maximum power of multipass automata. While we initially did not believe that all groups covered in the theorem were in \mathcal{D} it turns out that if we start with a group G in \mathcal{D} then the HNN extension H is indeed again in \mathcal{D} , that is, the class \mathcal{D} itself is closed under taking the mentioned HNN extensions. The paper concludes with a proof of this rather delicate algebraic fact.

So it turns out that starting with finitely generated virtually free groups, none of the closure properties for the class \mathcal{PG} which we establish take us outside the class \mathcal{D} . It now seems to us that Brough's Conjecture is probably true.

We point out that the complement of the word problem for a direct product of finitely many free groups is a context-free language and the class \mathcal{CG} of finitely generated groups whose word problem is the complement of a context-free language is closed under taking finite extensions and finitely generated subgroups [11]. Thus Brough's Conjecture also implies that $\mathcal{PG} \subsetneq \mathcal{CG}$.

There is still a good method for showing that languages are not poly-context-free, namely, one can use Parikh's theorem. We mention that Gorun [7] used properties of the Parikh map to prove that a certain bounded matrix language is not poly-context-free. Brough [2] developed a very detailed analysis of semi-linear sets which allows her to prove a hierarchy theorem for poly-context-free languages. For example, while the word problem for the free abelian group of rank k is an intersection of k context-free languages it is *not* an intersection of $k - 1$ context-free languages. Brough [2] also proves that none of the solvable Baumslag-Solitar groups $\langle b, t; tb^mt^{-1} = b^n \rangle$ where $0 < |m| < |n|$ are poly-context free.

The paper is organized as follows: we first define deterministic and nondeterministic multipass automata and study the closure properties of the corresponding classes of languages which they define. After proving the characterizations of the language classes, $\mathcal{DM} = \mathcal{BDC}$ and $\mathcal{NM} = \mathcal{PCF}$, we show that both classes are closed under interleaved products and left quotients by finite sets. We then study group word problems. Whether or not a group is in \mathcal{BDG} or in \mathcal{PG} is independent of the group presentation (Theorem 3.1) and both classes are closed under taking direct products (Corollary 3.2), finite extensions (Theorem 3.5) and finitely generated subgroups (Theorem 3.1). These easy results are to expected for a class of groups with word problems in a reasonable formal language class and were known for \mathcal{PG} and \mathcal{CG} . We then show (Theorem 4.1) that \mathcal{BDG} and \mathcal{PG} are closed under taking the HNN-extensions mentioned above. For $n \geq 2$ we construct a faithful representation of the Baumslag-Solitar group $BS(1, n^2)$ into $SL(2, \mathbb{Z}[\frac{1}{n}])$ showing that those matrix groups do not have poly-context-free word problem. In the last section we prove the rather delicate algebraic fact we alluded to above, namely that the class \mathcal{D} is closed under taking the mentioned HNN extensions.

2. MULTIPASS AUTOMATA AND CLOSURE PROPERTIES

2.1. Deterministic multipass automata. We first consider deterministic multipass automata. Let Σ be a finite input alphabet and let $k \geq 1$ be a positive integer. A *deterministic k -pass automaton* is a tuple

$$M = ([k], Q, \Sigma, \Gamma, \#, \delta, q_0, H_a, H_r)$$

where $[k] = \{1, 2, \dots, k\}$ is the *pass-counter* and, as usual, Q is a finite set of *states* and $q_0 \in Q$ is the *initial state*. The machine M starts its first pass at the beginning of the input tape in state q_0 with an empty stack. The finite set $\Gamma \supseteq \Sigma$ is the *stack alphabet*. We introduce the notation $\Sigma_\varepsilon := \Sigma \cup \{\varepsilon\}$ and $\Gamma_\varepsilon := \Gamma \cup \{\varepsilon\}$, where $\varepsilon \in \Sigma^*$ denotes the empty word.

The *end-marker* $\#$ is a letter not in Σ , and given any input word $w \in \Sigma^*$, the machine M will process $w\#$, the word w followed by $\#$. The distinct *halting states* H_a and H_r are not in Q .

We define the *transition function* δ by cases. When not reading the end-marker the transition function is a map

$$\delta: [k] \times Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow Q \times \Gamma^* \cup \emptyset$$

subject to the restrictions given below.

As usual, the interpretation of

$$\delta(j, q, \sigma, \gamma) = (q', \zeta)$$

where $j \in [k]$, $q, q' \in Q$, $\sigma \in \Sigma$, $\gamma \in \Gamma_\varepsilon$, and $\zeta \in \Gamma^*$, is that if the machine is on its j -th pass, in state q , and reading the letter σ on the input tape with γ on top of the stack, then the automaton changes state to q' , replaces γ by the word ζ and advances the input tape. Note that γ may be ε and the machine may continue working when the stack is empty. Indeed, our machines start each pass with empty stack.

The interpretation of

$$\delta(j, q, \varepsilon, \gamma) = (q', \zeta)$$

is, similarly, that if the machine is running its j -th pass, in state $q \in Q$, and reading a letter $\gamma \in \Gamma$ on top of the stack, then, independent of the input symbol being read, the automaton changes state to q' , and replaces γ by $\zeta \in \Gamma^*$. Such transitions are called ε -transitions and, in this case, the reading head does not advance on the tape.

Since we are considering *deterministic* machines, we require that if there is a transition $\delta(j, q, \varepsilon, \gamma)$ then $\delta(j, q, \sigma, \gamma)$ is empty for all $\sigma \in \Sigma$. Thus the machine has no choice between reading a letter and advancing the tape or making an ε transition. Also, for given $i \in \{1, 2, \dots, k\}$, $q \in Q$ and γ on top of the stack, the machine always has either a transition reading the input letter and advancing the tape or an ε -transition. Note also that the machine cannot make an ε -transition when the stack is empty.

When the automaton reads the end-marker \sharp on a nonfinal pass $j < k$ the transition function is a map

$$\delta: \{1, 2, \dots, k-1\} \times Q \times \{\sharp\} \times \Gamma_\varepsilon \rightarrow Q.$$

The interpretation of $\delta(j, q, \sharp, \gamma) = q'$ is that on reading the end-marker \sharp on finishing the j -th pass, in state $q \in Q$ with $\gamma \in \Gamma_\varepsilon$ on top of the stack, the automaton changes state to q' to begin the next pass. As part of the definition of the way the machine works, the reading head is automatically reset to the beginning of the input, the stack is emptied, and the pass-counter is advanced to $j+1$. Note that although the pass-counter is read by the machine, it functions automatically.

When reading the end-marker on the last pass the transition function is a map

$$(2.1) \quad \delta: \{k\} \times Q \times \{\sharp\} \times \Gamma_\varepsilon \rightarrow \{H_a, H_r\}.$$

On reading the end-marker on the last pass the machine must either halt in state H_a and accept or halt in state H_r and reject. Note that it is only on reading the end-marker on the last pass that the machine can go to either H_a or H_r and acceptance is thus completely determined by the last transition.

A deterministic k -pass automaton M *accepts* a word $w \in \Sigma^*$ if and only if when M is started in its initial state with an empty stack and with $w\sharp$ written on the input tape, then M halts in the accepting state H_a at the end of its k -th pass. We write $M \vdash w$ if M accepts w and denote by

$$L(M) := \{w \in \Sigma^* : M \vdash w\}$$

the *language accepted* by M .

A language which is accepted by a deterministic k -pass automaton is called a *deterministic k -pass language*. A *deterministic multipass language* is a language accepted by a deterministic k -pass automaton for some $k \geq 1$. We denote by \mathcal{DM} the class of all deterministic multipass languages.

2.2. Nondeterministic multipass automata. A *nondeterministic k -pass automaton* is a tuple

$$M = ([k], Q, \Sigma, \Gamma, \#, \delta, q_0, H_a, H_n)$$

where the notation is as before but the special state H_n is a *no decision* state. When not reading the end-marker the transition function is a map

$$\delta: [k] \times Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \rightarrow \mathcal{P}_f(Q \times \Gamma^*)$$

where $\mathcal{P}_f(Q \times \Gamma^*)$ denotes the collection of all finite subsets of $Q \times \Gamma^*$.

The interpretation of

$$\delta(j, q, \sigma, \gamma) = \{(q_1, \zeta_1), (q_2, \zeta_2), \dots, (q_r, \zeta_r)\}$$

is that if the machine M is on its j -th pass in state q and reads the letter σ on the input tape with the letter $\gamma \in \Gamma_\varepsilon$ on top of the stack, then the automaton can choose any of the pairs (q_i, ζ_i) and go to q_i as its next state, replace γ by the word ζ_i in the top of the stack, and advance the input tape.

The interpretation of

$$\delta(j, q, \varepsilon, \gamma) = \{(q_1, \zeta_1), (q_2, \zeta_2), \dots, (q_r, \zeta_r)\}$$

is, similarly, that if the machine is on its j -th pass in state q and with γ on top of the stack, then, independent of the input symbol being read, the automaton can choose any of the pairs (q_i, ζ_i) and go to q_i as its next state, replace γ by ζ_i , but the reading head does not advance on the tape. There may now be both transitions $\delta(j, q, \sigma, \gamma)$ and $\delta(j, q, \varepsilon, \gamma)$. If there is no transition from a given configuration the machine halts. We again require that the machine cannot make an ε -transition when the stack is empty.

On reading the end-marker on a nonfinal pass $j < k$ the transition function is a map

$$\delta: \{1, 2, \dots, k-1\} \times Q \times \{\#\} \times \Gamma_\varepsilon \rightarrow \mathcal{P}(Q).$$

The interpretation of $\delta(j, q, \#, \gamma) = Q_j \subseteq Q$ is that when the automaton reads the end-marker on a nonfinal pass $j < k$, in state q with γ on top of the stack, the automaton can change state to any $q' \in Q_j$ to begin the next pass. As before, the reading head is automatically reset to the beginning of the input, the stack is emptied, and the pass-counter is advanced to $j+1$.

On reading the end-marker on the final pass, the transition is a map

$$\delta: \{k\} \times Q \times \{\#\} \times \Gamma_\varepsilon \rightarrow \{H_a, H_n\}.$$

When the automaton reads the end-marker on the last pass in state $q \in Q$, with $\gamma \in \Gamma_\varepsilon$ on the top of the stack, the machine must either halt in state H_a and accept or halt in the no-decision state H_n . It is only on reading the end-marker on the last pass that the machine can go to either H_a or H_n .

A nondeterministic k -pass automaton M *accepts* a word $w \in \Sigma^*$ if and only if when M is started in its initial state with an empty stack and with $w\sharp$ written on the input tape, some possible computation of M on $w\sharp$ halts in the accepting state H_a . As in the deterministic case, we then write $M \vdash w$ if M accepts w , and denote by $L(M) := \{w \in \Sigma^* : M \vdash w\}$ the *language accepted* by M .

A language which is accepted by a nondeterministic k -pass automaton is called a *nondeterministic k -pass language*. A *nondeterministic multipass language* is a language accepted by a nondeterministic k -pass automaton for some $k \geq 1$. We denote by \mathcal{NM} the class of all nondeterministic multipass languages. Of course, $\mathcal{DM} \subseteq \mathcal{NM}$.

2.3. Properties of deterministic and nondeterministic multipass languages. A basic fact about the class of languages accepted by deterministic pushdown automata is that the class is closed under complementation. The only obstacle to proving this is that at some point the automaton might go into an unbounded sequence of ε -transitions. Call a deterministic pushdown automaton *complete* if it always reads its entire input. Similarly, we call a multipass automaton *complete* if it always reads the end-marker on every pass. A basic lemma [12, Lemma 10.3] shows that for any deterministic pushdown automaton there is a complete pushdown automaton accepting the same language. The proof for deterministic multipass automata is essentially the same but even easier since we do not have to worry about final states.

Lemma 2.1. *For every deterministic multipass automaton there is a complete multipass automaton accepting the same language.*

Proof. Let $M = ([k], Q, \Sigma, \Gamma, \#, \delta, q_0, H_a, H_r)$ be a deterministic k -pass automaton. We construct a complete k -pass automaton $M' = ([k], Q', \Sigma, \Gamma, \#, \delta', q_0, H_a, H_r)$ as follows. $Q' := Q \sqcup \{r\}$, where r is a new rejecting state. The machine M' basically works as M so that for the new transition map δ' we have

$$\delta'(j, q, \sigma, \gamma) = \delta(j, q, \sigma, \gamma)$$

and

$$\delta'(j, q, \#, \gamma) = \delta(j, q, \#, \gamma)$$

for all $j \in [k]$, $q \in Q$, $\sigma \in \Sigma$, and $\gamma \in \Gamma$. However, if on some pass $j \in [k]$ and in some state $q \in Q$ with a letter γ on top of the stack, M would start an unbounded sequence of ε -transitions without erasing that occurrence of γ then M' will instead enter the rejecting state r where it remains and then simply reads all input letters for all remaining passes until it reads the end-marker on the last pass and then rejects.

For example, if we have the sequence of transitions

$$\delta(j, q_i, \varepsilon, \gamma_i) = (q_{i+1}, \zeta_{i+1}),$$

where $q_i \in Q$ ($q_0 = q$), $\gamma_i \in \Gamma$ ($\gamma_0 = \gamma$), $\zeta_{i+1} \in \Gamma^*$, $i \in \mathbb{N}$, and there exists $i_0 > 0$ such that $q_{i_0} = q$ and $\zeta_{i_0} = \zeta'\gamma$ for some $\zeta' \in \Gamma^*$, we would then set

$$\begin{aligned}\delta'(j', q, \varepsilon, \gamma) &:= (r, \gamma) \\ \delta'(j', r, \sigma, \gamma) &:= (r, \gamma) \\ \delta'(j', r, \#, \gamma) &:= \begin{cases} r & \text{if } j' < k \\ H_r & \text{if } j' = k \end{cases}\end{aligned}$$

for all $j' = j, j+1, \dots, k$, $q \in Q$, $\sigma \in \Sigma$, and $\gamma \in \Gamma \cup \{\varepsilon\}$.

It is clear that $L(M') = L(M)$. This construction can be made effective but we only need the stated result. \square

Proposition 2.2. *The class \mathcal{DM} of deterministic multipass languages is closed under complementation.*

Proof. Let $L \subseteq \Sigma^*$ be a deterministic multipass language and let M be a complete deterministic multipass automaton accepting L . The complement $\neg L := \Sigma^* \setminus L$ of L is accepted by the multipass automaton M^\neg which is the same as M except that it does the opposite of what M does on reading the end-marker on the final pass, that is, it exchanges the accepting and rejecting states. \square

Proposition 2.3. *The class of deterministic context-free languages coincides with the class of deterministic 1-pass languages and the class of context-free languages coincides with the class of nondeterministic 1-pass languages.*

Proof. This result is intuitively clear but we need to check some details. Let $L \subseteq \Sigma^*$ be a deterministic context-free language and let $M = (Q, \Sigma, \Gamma, \delta, q_0, F, Z_0)$ be a complete deterministic pushdown automaton accepting L by entering a final state. Note that M has start symbol Z_0 on the stack and scans the whole input. Then L is accepted by the deterministic 1-pass automaton

$$M' = ([1], Q', \Sigma, \Gamma, \#, \delta', q'_0, H_a, H_r)$$

defined as follows. The set of states of M' is $Q' := Q \sqcup \{q'_0\}$. Then, on reading the first letter of the input, M' adds Z_0 to the stack followed by whatever M would add to the stack, so

$$\delta'(1, q'_0, \sigma, \varepsilon) := (q, Z_0\zeta) \quad \text{if} \quad \delta(q_0, \sigma, Z_0) = (q, \zeta)$$

for all $\sigma \in \Sigma$, where $q \in Q$ and $\zeta \in \Gamma^*$. Then, M' simulates M :

$$\delta'(1, q, \sigma, \gamma) = \delta(q, \sigma, \gamma)$$

and, finally,

$$\delta'(1, q, \#, \gamma) = \begin{cases} H_a & \text{if } q \in F \\ H_r & \text{otherwise,} \end{cases}$$

for all $q \in Q$, $\sigma \in \Sigma$, and $\gamma \in \Gamma$.

Thus $L = L(M')$ is a deterministic 1-pass language. The proof is essentially the same for the nondeterministic case. However we have to replace Q' by $Q'' := Q \sqcup \overline{Q} \sqcup \{q'_0, r\}$,

where \overline{Q} is a disjoint copy of Q . Then, on a sequence of ε -transitions of M , the 1-pass automaton M' remembers if M ever entered a final state in F during the sequence (this is achieved by moving from state q to its “accepting” copy $\overline{q} \in \overline{Q}$). Then, if M' reads \sharp as its next letter it accepts if M did enter a final state or rejects if not. Also, if M emptied its stack without entering a final state on its current input, then M' goes to a rejecting state $r \in Q'$ in which it remains and simply reads the remaining input until it encounters \sharp and then rejects.

Conversely, let L be a deterministic 1-pass language and let

$$M = ([1], Q, \Sigma, \Gamma, \sharp, \delta, q_0, H_a, H_r)$$

be a deterministic 1-pass automaton accepting L which, by Lemma 2.1, we may suppose being complete. We now construct a (complete) deterministic pushdown automaton $M' = (Q', \Sigma, \Gamma', \delta', q_0, F, Z_0)$ which accepts L as follows. $Q' := \{q'_0\} \sqcup (Q \times \Gamma_\varepsilon)$ and $\Gamma' := \Gamma \sqcup \{Z_0\}$.

M' starts with a new beginning symbol $Z_0 \notin \Gamma$, which it will never erase and which it treats it the same way as M treats an empty stack, and then simulates M : thus

$$\delta'(q_0, \sigma, Z_0) := ((q', \gamma'), \zeta\gamma') \quad \text{if} \quad \delta(1, q_0, \sigma, \varepsilon) = (q', \zeta\gamma')$$

and

$$\delta'((q, \gamma''), \sigma, \gamma) := ((q', \gamma'''), \zeta\gamma''') \quad \text{if} \quad \delta(1, q, \sigma, \gamma) = (q', \zeta\gamma'''),$$

where $q' \in Q$, $\gamma', \gamma''' \in \Gamma_\varepsilon$, and $\zeta \in \Gamma^*$, for all $q \in Q$, $\sigma \in \Sigma \cup \{\varepsilon\}$ and $\gamma, \gamma'' \in \Gamma$. The set F of final states of M' is defined by

$$F := \{(q, \gamma) \in Q' : \delta(1, q, \sharp, \gamma) = H_a\}.$$

Then, it is then clear that M' accepts L by final state, so that $L = L(M')$. Again, the proof is essentially the same for the nondeterministic case. M' will now have two copies of the state set of M , one accepting and one rejecting and simulate M . However, on ε -transitions of M , the pushdown automaton M' always goes to a non-accepting copy of the corresponding state. On a transition advancing the tape, if M would accept when reading \sharp as the next letter, M' goes to the accepting copy of the state of M and otherwise to the non-accepting copy. \square

Proposition 2.4. *Both the classes of deterministic and nondeterministic multipass languages are closed under union and intersection.*

Proof. Let $L_i \subseteq \Sigma^*$ be accepted by a deterministic k_i -pass automaton M_i for $i = 1, 2$. We suppose that M_1 and M_2 have disjoint state sets Q_1 and Q_2 , respectively, and set $k := k_1 + k_2$.

We construct a k -pass automaton M accepting the union $L_1 \cup L_2$ as follows. M will have state set $Q_1 \cup Q_2 \cup \{a\}$, where a is a special accepting state. On the first k_1 passes M simulates M_1 . On reading the end-marker \sharp at the end of pass k_1 , if M_1 would accept then M goes to the accepting state a in which it remains while reading the input for the remaining k_2 passes, and then accepts on the k -th pass. If M_1 would reject then M goes to the first initial state of M_2 and then simulates M_2 on the next k_2 passes. On reading

the end-marker \sharp at the end of the k th pass, if M_2 would accept then M accepts, otherwise M rejects. This shows that the language accepted by M is $L_1 \cup L_2$.

We can similarly construct a k -pass automaton M accepting the intersection $L_1 \cap L_2$ as follows. The argument is essentially the same as before, but now M has state set $Q_1 \cup Q_2 \cup \{r\}$ where r is a rejecting state. At the end of pass k_1 , if M_1 would reject, then M goes to the rejecting state r in which it remains while reading the input for the remaining k_2 passes, and then rejects on the final pass. On the other hand, if M_1 accepts then, for the remaining k_2 passes, M just simulates M_2 and, in particular, does whatever M_2 at the end of the final pass. It is then clear that the language accepted by M is $L_1 \cap L_2$.

We now consider the nondeterministic case. So, let $L_i \subseteq \Sigma^*$ be accepted by nondeterministic k_i -pass automaton M_i for $i = 1, 2$. We set $k := k_1 + k_2$.

Then the k -pass nondeterministic automaton M accepting $L_1 \cup L_2$ works as follows. M simulates M_1 on the first k_1 passes and if the computation of M_1 would accept, then, as before, M goes to an accepting state in which it remains while reading the input for the remaining k_2 passes, and then accepts on the k -th pass. If not, then M goes to the first initial state of M_2 and then simulates M_2 on the next k_2 passes.

Similarly, for the intersection, M accepts exactly if computations of M_1 and M_2 both accept. \square

From the previous results we immediately deduce the following.

Corollary 2.5. *The class \mathcal{DM} of languages accepted by deterministic multipass automata contains the Boolean closure of the class of deterministic context-free languages. The class \mathcal{NM} of languages accepted by nondeterministic multipass automata contains the closure of context-free languages under union and intersection.*

We now turn to proving the converse of Corollary 2.5.

Theorem 2.6. *The class \mathcal{DM} of deterministic multipass languages is the Boolean closure of the class of deterministic context-free languages and the class \mathcal{NM} of nondeterministic multipass languages is the class \mathcal{PCF} of poly-context-free languages.*

Proof. Let $M = ([k], Q, \Sigma, \Gamma, \sharp, \delta, q_0, H_a, H_r)$ be a k -pass automaton, which we assume to be complete if M is deterministic. If $w \in \Sigma^*$ and $M \vdash w$, define an M -accepting profile of w as an ordered sequence of k triples:

$$p(w) = ((q_{1,0}, \gamma_1, q_{1,1}), (q_{2,0}, \gamma_2, q_{2,1}), \dots, (q_{k,0}, \gamma_k, q_{k,1}))$$

where $q_{i,0} \in Q$ is the initial state beginning the i -th pass and $q_{i,1} \in Q$ (respectively $\gamma_i \in \Gamma_\varepsilon$) is the control state (respectively the top letter of the stack) on reading the end-marker on the i -th pass in an accepting computation of M on w (so that $\delta(k, q_{k,1}, \sharp, \gamma_k) = H_a$). Note that if M is deterministic there is of course only one such profile. Let $P(M) = \{p(w) : w \in L(M)\}$ denote the set of all such M -accepting profiles, and note that $P(M)$ is finite since $P(M) \subseteq (Q \times \Gamma_\varepsilon \times Q)^k$.

Let $P(M) = \{p_1, p_2, \dots, p_t\}$ and suppose that

$$p_i = ((q_{1,0}^i, \gamma_1^i, q_{1,1}^i), (q_{2,0}^i, \gamma_2^i, q_{2,1}^i), \dots, (q_{k,0}^i, \gamma_k^i, q_{k,1}^i))$$

for $i = 1, 2, \dots, t$. Let $L_{i,j}$ be the language consisting of all words $w \in \Sigma^*$ accepted by the 1-pass automaton $M_{i,j}$ which, starting in state $q_{j,0}^i$, simulates M on reading w^\sharp on the j -th pass, and then accepts exactly when reading the end-marker in state $q_{j,1}^i$ with γ_j^i on top of the stack. Note that for $w \in \Sigma^*$ one has $p(w) = p_i$ if and only if $w \in \bigcap_{j=1}^k L_{i,j}$.

As a consequence,

$$(2.2) \quad L(M) = \bigcup_{i=1}^t \bigcap_{j=1}^k L_{i,j}.$$

Since De Morgan's laws allow us to move negations inside unions and intersections in a Boolean expression, the Boolean closure of the class of deterministic context-free languages is the same as its closure under union and intersection. Since the class of arbitrary context-free languages is closed under unions, we can distribute unions past intersections and the closure of the class of context-free languages under unions and intersections is the same as its closure under intersections. This completes the proof of the theorem. \square

Recall [10] that a gsm, a *generalized sequential machine*,

$$S = (Q, \Sigma, \Delta, \lambda, q_0)$$

is a deterministic finite automaton with output. On reading a letter $\sigma \in \Sigma$ in state q , the machine outputs a word $\lambda(q, \sigma) \in \Delta^*$. Since S is deterministic it defines a map $g: \Sigma^* \rightarrow \Delta^*$. The convention is that $g(\varepsilon) = \varepsilon$. For example, a homomorphism $\phi: \Sigma^* \rightarrow \Delta^*$ can be defined by a one-state gsm: on reading a letter σ the gsm outputs $\phi(\sigma)$. It is well-known that both deterministic and nondeterministic context-free languages are closed under inverse gsm mappings. (If L is context-free, incorporate S into a pushdown automaton M accepting L , and when S would output u simulate M on reading u .) Since inverse functions commute with Boolean operations we have:

Proposition 2.7. *The classes \mathcal{DM} and \mathcal{PCF} are both closed under inverse gsm mappings. In particular, \mathcal{DM} and \mathcal{PCF} are both closed under inverse monoid homomorphisms. \square*

Definition 2.8. Let $\Sigma_i, i = 1, 2, \dots, r$ be finite alphabets and let $L_i \subseteq \Sigma_i^*, i = 1, 2, \dots, r$. Let $\Sigma = \bigcup_{i=1}^r \Sigma_i$ and denote by $\pi_i: \Sigma^* \rightarrow \Sigma_i^*$ the monoid homomorphism defined by setting

$$\pi_i(a) = \begin{cases} a & \text{if } a \in \Sigma_i \\ \varepsilon & \text{otherwise.} \end{cases}$$

We call the language

$$L = \{w \in \Sigma^* : \pi_i(w) \in L_i, i = 1, 2, \dots, r\}$$

the *interleaved product* of the languages L_i .

Note that in the definition above there is no hypothesis on how the Σ_i overlap. If the alphabets are all disjoint then L is the *shuffle product* of the L_i . On the other hand, if the alphabets are all the same then L is the intersection of the L_i . There does not seem to be a standard name if the overlap of the alphabets is arbitrary.

Proposition 2.9. *The classes \mathcal{BDC} and \mathcal{PCF} are closed under interleaved product.*

Proof. With the notation above, the interleaved product of the languages L_i is

$$\bigcap_{i=1}^r \pi_i^{-1}(L_i)$$

The statement then follows from Proposition 2.7 and Proposition 2.4. \square

Recall [10] that if K and L are subsets of Σ^* then the *left quotient* of L by K is the language

$$K^{-1}L = \{w \in \Sigma^* : \exists u \in K \text{ such that } uw \in L\}.$$

Proposition 2.10. *The classes \mathcal{BDC} and \mathcal{PCF} are closed under left quotients by finite sets.*

Proof. Let $L \subseteq \Sigma^*$ be in \mathcal{BDC} (respectively \mathcal{PCF}) and let K be a finite subset of Σ^* . Since $K^{-1}L = \bigcup_{u \in K} \{u\}^{-1}L$ and the classes \mathcal{BDC} and \mathcal{PCF} are closed under finite unions (cf. Proposition 2.4), we may suppose that $K = \{u\}$.

We then define a two-state generalized sequential machine S defining a mapping $g: \Sigma^* \rightarrow \Sigma^*$ as follows. On reading a letter σ in its initial state, S outputs $u\sigma$ and then goes to its second state, in which it will stay. In its second state S simply outputs σ on reading σ .

Since $u \in L \Leftrightarrow \varepsilon \in \{u\}^{-1}L$ and $\varepsilon \in L \Leftrightarrow \varepsilon \in g^{-1}(L)$ (recall that $g(\varepsilon) = \varepsilon$), it is then clear that

$$\{u\}^{-1}L = \begin{cases} g^{-1}(L) \cup \{\varepsilon\} & \text{if } u \in L \text{ and } \varepsilon \notin L \\ g^{-1}(L) \setminus \{\varepsilon\} & \text{if } u \notin L \text{ and } \varepsilon \in L \\ g^{-1}(L) & \text{otherwise.} \end{cases}$$

Now, since $\{\varepsilon\}$ and its complement are regular languages in Σ^* and the classes \mathcal{BDC} and \mathcal{PCF} are closed under finite unions/intersections (cf. Proposition 2.4), the result then follows from Proposition 2.7. \square

3. GROUP WORD PROBLEMS

Let $G = \langle X; R \rangle$ be a finitely generated group presentation. We denote by $\Sigma := X \cup X^{-1}$ the associated *group alphabet*. Then the *Word Problem* of G (cf. [1, 4]), relative to the given presentation, is the language

$$\text{WP}(G : X; R) := \{w \in \Sigma^* : w = 1 \text{ in } G\} \subseteq \Sigma^*$$

where for $w, w' \in \Sigma^*$ we write “ $w = w'$ in G ” provided $\pi(w) = \pi(w')$, where $\pi: \Sigma^* \rightarrow G$ denotes the canonical monoid epimorphism.

Theorem 3.1. *Let $G = \langle X; R \rangle$ be a finitely generated group. Then whether or not the associated word problem $\text{WP}(G : X; R)$ is in \mathcal{BDC} (respectively \mathcal{PCF}) is independent of the given presentation. Moreover, if the word problem of G is in \mathcal{BDC} (respectively \mathcal{PCF}), then every finitely generated subgroup of G also has word problem in \mathcal{BDC} (respectively \mathcal{PCF}).*

Proof. Suppose that the word problem $\text{WP}(G : X; R)$ is a deterministic (respectively nondeterministic) multipass language. Let $H = \langle Y; S \rangle$ be a finitely generated group and suppose that there is an injective group homomorphism $\phi: H \rightarrow G$. Let $\Sigma = X \cup X^{-1}$ and $Z = Y \cup Y^{-1}$ denote the corresponding group alphabets. For each $y \in Y$ let $w(y) \in \Sigma^*$ be a word representing the group element $\phi(y) \in G$ and consider the unique monoid homomorphism $\psi: Z^* \rightarrow \Sigma^*$ satisfying $\psi(y) = w(y)$ and $\psi(y^{-1}) = w(y)^{-1}$ for all $y \in Y$. Let $w \in Z^*$. Then $w \in \text{WP}(H : Y; S)$ if and only if $\psi(w) \in \text{WP}(G : X; R)$, that is, $\text{WP}(H : Y; S) = \psi^{-1}(\text{WP}(G : X; R))$. From Proposition 2.7 we then deduce that $\text{WP}(H : Y; S)$ is deterministic (respectively nondeterministic) multipass. This proves the second part of the statement. Taking H isomorphic to G gives the first part of the statement. \square

We denote by \mathcal{BDG} the class of all finitely generated groups having a deterministic multipass word problem and by \mathcal{PG} the class of all finitely generated groups having a nondeterministic multipass, equivalently poly-context-free, word problem.

Since the word problem of the direct product of two groups given by presentations on disjoint sets of generators is the shuffle product of the word problems of the two groups, from Proposition 2.9 we immediately deduce:

Corollary 3.2. *The classes \mathcal{BDG} and \mathcal{PG} are closed under finite direct products. That is, if two groups G_1 and G_2 are in \mathcal{BDG} (respectively \mathcal{PG}) then $G_1 \times G_2$ is in \mathcal{BDG} (respectively \mathcal{PG}).*

Stallings' example [19] of a finitely generated subgroup of $F_2 \times F_2$ which is not finitely presented is the kernel of the homomorphism

$$F_2 \times F_2 \rightarrow \mathbb{Z}$$

defined by mapping every free generator of each factor of $F_2 \times F_2$ to a fixed generator of the infinite cyclic group \mathbb{Z} . Thus \mathcal{BDG} and \mathcal{PG} contain groups which are *not* finitely presentable.

Corollary 3.3. *All finitely generated abelian groups are in \mathcal{BDG} .*

Theorem 3.4. *The classes \mathcal{BDG} and \mathcal{PG} are closed under quotients by finite normal subgroups.*

Proof. Let $G = \langle X; R \rangle$ be in \mathcal{BDG} (respectively \mathcal{PG}) and let $N = \{1 = n_1, n_2, \dots, n_r\}$ be a finite normal subgroup of G . Then the quotient group $H = G/N$ admits the presentation

$$(3.1) \quad H = \langle X; R \cup N \rangle.$$

Let $\Sigma = X \cup X^{-1}$ denote the group alphabet for both G and H and let W be the word problem of G . If $w \in \Sigma^*$ then $w = 1$ in H if and only if $w \in N$ in G . Let K be the finite set of *inverses* of elements of N . Then $w \in N$ if and only if $w \in K^{-1}W \cup W$, and the result follows from Proposition 2.10 and Proposition 2.4. \square

It is known that both the class of finitely generated groups with context-free co-word problem [11] and the class of groups with poly-context-free word problem [2] are closed under finite extensions.

Theorem 3.5. *The classes \mathcal{BDG} and \mathcal{PG} are closed under finite extensions.*

Proof. Let $G = \langle X; R \rangle$ be a finitely generated group with a subgroup $H = \langle Y; S \rangle$ of finite index which is in \mathcal{BDG} (respectively \mathcal{PG}). Note that H must be finitely generated. Let $\{1 = c_1, c_2, \dots, c_k\}$ be a set of representatives of the left cosets of H in G . For each coset representative c_i and each $x \in X^{\pm 1}$ there exist a unique representative c_j , $j = j(i, x)$, and a word $u_{i,x}$ in the generators Y of H such that $c_i x = c_j u_{i,x}$. Let M be a deterministic (respectively nondeterministic) multipass automaton accepting the word problem $\text{WP}(H : Y; S)$ of H .

Define a multipass automaton \widehat{M} accepting the word problem $\text{WP}(G : X; R)$ of G as follows. The state set of \widehat{M} will keep track of the coset representative of the word read so far. If the coset is c_i and \widehat{M} reads a letter $x \in X^{\pm 1}$, then \widehat{M} changes the current coset to c_j , $j = j(i, x)$, and simulates M on reading $u_{i,x}$. At the end of its final pass, \widehat{M} accepts exactly if the current coset is c_1 and M would accept. \square

We can immediately use the above theorem to prove the following:

Theorem 3.6. *Let G_1 and G_2 be two finitely generated groups in \mathcal{BDG} (respectively \mathcal{PG}). Suppose that G_2 acts on G_1 by a finite group of automorphisms. Then the corresponding semi-direct product $G_1 \rtimes G_2$ is also in \mathcal{BDG} (respectively \mathcal{PG}).*

Proof. Since G_2 acts on G_1 by a finite group of automorphisms, the subgroup $H \leq G_2$ which fixes all the elements of G_1 has finite index in G_2 and therefore is in \mathcal{BDG} (respectively \mathcal{PG}) by Theorem 3.1. As a consequence, the subgroup $G_1 \rtimes H$ is just $G_1 \times H$ and has finite index in $G_1 \rtimes G_2$ and the statement follows from Corollary 3.2 and Theorem 3.5. \square

If $G = \langle X; R \rangle$ is a finitely generated group and φ is an automorphism of G then the mapping torus of φ is the HNN-extension

$$\langle G, t; txt^{-1} = \varphi(x), x \in X \rangle.$$

Corollary 3.7. *Let G be a group in \mathcal{BDG} (respectively \mathcal{PG}) and suppose that ϕ is an automorphism of G of finite order. Then the mapping torus of ϕ is also in \mathcal{BDG} (respectively \mathcal{PG}).*

4. HNN-EXTENSIONS AND DOUBLES

We have seen that the classes \mathcal{BDG} and \mathcal{PG} are closed under taking mapping tori of automorphisms of finite order. In this section we prove a much more general result.

Theorem 4.1. *Let $G = \langle X; R \rangle$ in \mathcal{BDG} (respectively \mathcal{PG}) and S a subgroup of G of finite index. Suppose also that $\varphi: G \rightarrow G$ is an automorphism of finite order such that $\varphi(S) = S$. Then the HNN-extension*

$$H = \langle G, t; tst^{-1} = \varphi(s), s \in S \rangle$$

is again in \mathcal{BDG} (respectively \mathcal{PG}).

Proof. First recall that a subgroup S of finite index contains a characteristic subgroup C of finite index. Let $G/C =: K = \{1_K = k_1, k_2, \dots, k_r\}$ be the corresponding finite quotient group and let $\psi: G \rightarrow K$ be the associated epimorphism. Since C is characteristic, φ induces an automorphism $\overline{\varphi}: K \rightarrow K$. Finally, let $J := \psi(S) \leq K$ denote the image of S in K . By Britton's Lemma, if a word $w = 1$ in H , then all occurrences of the stable letter t must cancel by successive t -reductions. Now for $u \in G$ we have $t^\epsilon u t^{-\epsilon} = \varphi^\epsilon(u)$ (where $\epsilon = \pm 1$) if and only if $u \in S$, or equivalently, if and only if $\psi(u) \in J$.

Let $\Sigma(X) := X \cup X^{-1}$ be the group alphabet for the base group G and let M be a multipass automaton accepting $\text{WP}(G : X; R) \subseteq \Sigma(X)^*$, say with stack alphabet Δ .

We now describe a new multipass automaton \widehat{M} with input alphabet $\Sigma := \Sigma(X) \cup \{t, t^{-1}\}$ and stack alphabet $\Delta \cup (\{t, t^{-1}\} \times K)$, which will accept the word problem of H .

An input word has the form $w = ut^{\epsilon_1}u_1t^{\epsilon_2} \dots u_{n-1}t^{\epsilon_n}u_n$, where $u, u_i \in \Sigma(X)^*$ and $\epsilon_i = \pm 1$ for $i = 1, 2, \dots, n$.

The first pass of \widehat{M} is to decide whether or not all the t 's cancel. Let $p \geq 1$ denote the order of the automorphism φ . As \widehat{M} proceeds from left to right, it will count the algebraic sums $m_i := \epsilon_1 + \epsilon_2 + \dots + \epsilon_i \bmod p$ for $i = 1, 2, \dots, n$.

To begin, \widehat{M} ignores u , that is, any letters from the base group until the first t^{ϵ_1} , and then adds $(t^{\epsilon_1}, 1_K)$ to the stack and counts $\epsilon_1 \bmod p$. Next, \widehat{M} keeps track of the element $k := \psi(\varphi^{\epsilon_1}(u_1)) = \overline{\varphi}^{\epsilon_1}(\psi(u_1)) \in K$. There are now two cases. If $k \notin J$ or if $\epsilon_2 = \epsilon_1$, then \widehat{M} replaces $(t^{\epsilon_1}, 1_K)$ with $(t^{\epsilon_1}, k)(t^{\epsilon_2}, 1_K)$ on the stack. If $k \in J$ and $\epsilon_2 = -\epsilon_1$ then \widehat{M} erases $(t^{\epsilon_1}, 1_K)$ from the stack which then remains empty. In either case, \widehat{M} now counts $m_2 = \epsilon_1 + \epsilon_2 \bmod p$.

Suppose that the machine has just finished processing the i -th t -symbol, so that the count is $m_i = (\epsilon_1 + \epsilon_2 + \dots + \epsilon_i) \bmod p$, and the stack is either empty or the top symbol on the stack is (t^ϵ, k) . If the stack is empty, then as before, \widehat{M} reads the base letters until the next t^η -symbol is encountered and then puts $(t^\eta, 1_K)$ on the stack. If there is (t^ϵ, k) on top of the stack, the machine keeps track of the the image $k' := \psi(\varphi^{m_i}(u)) = \overline{\varphi}^{m_i}(\psi(u)) \in K$ of the product u of generators in the base from that point until the next occurrence, say t^η , of a t -symbol.

If $\eta = -\epsilon$ and $kk' \in J$, then \widehat{M} does the following. It removes (t^ϵ, k) on top of the stack and, if the stack is empty then \widehat{M} continues processing the input; if there is now a (t^β, k'') on the top of the stack, then \widehat{M} replaces that symbol by $(t^\beta, k''kk')$.

On the other hand, if $\eta = \epsilon$ or $kk' \notin J$ then \widehat{M} erases (t^ϵ, k) on top of the stack and replaces it by $(t^\epsilon, kk')(t^\eta, 1_K)$.

On reading the end-marker, all the t 's have cancelled if and only if the stack is empty. If the stack is nonempty \widehat{M} goes to a rejecting state that will reject on reading the end-marker on the final pass. If the stack is empty, since $t^\epsilon s t^{-\epsilon} = \varphi^\epsilon(s)$ in G for all $s \in S$ and $\epsilon = \pm 1$, then after cancelling all t -symbols, we have $w = u\varphi^{m_1}(u_1)\varphi^{m_2}(u_2) \dots \varphi^{m_n}(u_n)$ in G . Then \widehat{M} ignores t -symbols and just simulates M to check that the word $u\varphi^{m_1}(u_1)\varphi^{m_2}(u_2) \dots \varphi^{m_n}(u_n)$ is equal to the identity in G . \square

Definition 4.2. Let G (respectively \overline{G}) be a group and let $S \leq G$ be a subgroup. Let also $\varphi: G \rightarrow G$ be an automorphism of finite order such that $\varphi(S) = S$. Suppose there exists an isomorphism $g \mapsto \overline{g}$ of G onto \overline{G} and denote by $\overline{S} \leq \overline{G}$ the image of the subgroup S . Consider the free product with amalgamation

$$D := \langle G * \overline{G}; \varphi(s) = \overline{s}, s \in S \rangle.$$

If φ is the identity of G , then D is called a *double* of G over the subgroup S . In general, if φ is not the identity, then D is called a *double* of G over the subgroup S *twisted* by the automorphism φ .

Note that for a double we require that the amalgamation is induced by the global isomorphism between G and \overline{G} .

Theorem 4.3. Let G be a group in \mathcal{BDG} (respectively \mathcal{PG}) and suppose that S is a subgroup of finite index in G . Let also $\varphi: G \rightarrow G$ be an automorphism of finite order such that $\varphi(S) = S$. Then $D := \langle G * \overline{G}; \varphi(s) = \overline{s}, s \in S \rangle$ is again in \mathcal{BDG} (respectively \mathcal{PG}).

Proof. Note that D embeds in the HNN-extension

$$H = \langle G, t; tst^{-1} = \varphi(s), s \in S \rangle$$

via the map defined by $g \mapsto g$ and $\overline{g} \mapsto tg^{-1}t^{-1}$ for all $g \in G$ and $\overline{g} \in \overline{G}$. This is a general fact that does not require any particular hypothesis on G or S . But if G is in \mathcal{BDG} (respectively \mathcal{PG}) and S is a subgroup of finite index, then H is in \mathcal{BDG} (respectively \mathcal{PG}) by Theorem 4.1. Since D is isomorphic to a finitely generated subgroup of H , we deduce from Theorem 3.1 that D is also in \mathcal{BDG} (respectively \mathcal{PG}). \square

5. PARIKH'S THEOREM AND SOME EXAMPLES OF GROUPS NOT IN \mathcal{PG}

We start by recalling some known preliminary facts.

Let $\Sigma = \{a_1, a_2, \dots, a_r\}$. Then Σ^* is the *free monoid of rank r* . The free *commutative* monoid of rank r is \mathbb{N}^r with vector addition. The natural *abelianization map* $\pi: \Sigma^* \rightarrow \mathbb{N}^r$ is defined by $\pi(a_i) = \mathbf{e}_i$, where the vector $\mathbf{e}_i \in \mathbb{N}^r$ has 1 in the i -th coordinate and 0 elsewhere, for all $i = 1, 2, \dots, r$. Note that if $w \in \Sigma^*$ then the i -th coordinate of $\pi(w)$ equals the number of occurrences of the letter a_i in w . In formal language theory the abelianization map is called the *Parikh mapping* because of a remarkable theorem of Parikh [16, 17, 10] which we now review.

Definition 5.1. A *linear* subset of \mathbb{N}^r is a set of the form

$$(5.1) \quad S = S(\mathbf{v}_0; \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m) := \{\mathbf{v}_0 + n_1\mathbf{v}_1 + \dots + n_m\mathbf{v}_m : n_1, n_2, \dots, n_m \in \mathbb{N}\}$$

where $m \geq 0$ and $\mathbf{v}_i \in \mathbb{N}^r$ for all $i = 0, 1, \dots, m$. A *semi-linear* subset of \mathbb{N}^r is a finite union of linear subsets of \mathbb{N}^r .

Theorem 5.2 (Parikh). Let $L \subseteq \Sigma^*$ be a context-free language. Then $\pi(L) \subseteq \mathbb{N}^r$ is semi-linear and there is a regular language $R \subseteq \Sigma^*$ such that $\pi(R) = \pi(L)$.

Thus abelianizations cannot distinguish between regular languages and context-free languages! It is this property that one can use to show that certain groups are not in \mathcal{PG} . A clear proof of Parikh's theorem is found in [17].

It is easy to show that any semi-linear subset of \mathbb{N}^r is the image of a regular language under the abelization map π . Indeed, in the notation of Definition 5.1, let $\mathbf{v}_i = (v_{i,1}v_{i,2}, \dots, v_{i,r})$ for $i = 0, 1, \dots, m$. Consider the regular language

$$V := (a_1^{v_{0,1}} a_2^{v_{0,2}} \cdots a_r^{v_{0,r}}) (a_1^{v_{1,1}} a_2^{v_{1,2}} \cdots a_r^{v_{1,r}})^* \cdots (a_1^{v_{m,1}} a_2^{v_{m,2}} \cdots a_r^{v_{m,r}})^*.$$

Looking back to (5.1), it is clear that $\pi(V) = S$. Since the class of regular languages is closed under union, any semi-linear set is the image of a regular language under π .

It is well-known [10] that a Boolean combination of semi-linear sets is again semi-linear. Indeed, the semi-linear sets are exactly the sets definable in Presburger arithmetic [6].

Let $L = \bigcap_{j=1}^q L_j$ be a poly-context-free language. Now *suppose* that the restriction to $\bigcup_{j=1}^q L_j$ of the Parikh map π is injective. Then

$$\pi(L) = \pi\left(\bigcap_{j=1}^q L_j\right) = \bigcap_{j=1}^q \pi(L_j)$$

and thus $\pi(L)$ is semi-linear. This gives a general strategy for showing that some word problems cannot be poly-context-free.

Brough [2] proved the general result that none of the Baumslag-Solitar groups

$$BS(1, n) = \langle b, t; tb^m t^{-1} = b^n \rangle$$

where $1 \leq |m| < |n|$ are in \mathcal{PG} . We illustrate the technique outlined above by giving a simple proof of the following:

Proposition 5.3 (Brough). *None of the Baumslag-Solitar groups*

$$BS(1, n) = \langle b, t; tbt^{-1} = b^n \rangle, n \geq 2$$

have poly-context-free word problem.

Proof. Fix $n \geq 2$. Let $\Sigma = \{b, b^{-1}, t, t^{-1}\}$ be the associated group alphabet and let $W \subseteq \Sigma^*$ be the word problem for the Baumslag-Solitar group $BS(1, n)$. We write t^+ (respectively t^- , respectively b^-) to denote the set of all words of the form t^s (respectively t^{-s} , respectively b^{-s}), where $s \in \mathbb{N}$. We then consider the regular language

$$R := t^+ b t^- b^- \subseteq \Sigma^*.$$

Observe that $W' := W \cap R$ is exactly $\{t^s b t^{-s} b^{-n^s} : s \geq 1\}$.

Suppose by contradiction that the word problem W for $BS(1, n)$ is poly-context-free. Then $W = \bigcap_{j=1}^q L_j$, where $L_j \subseteq \Sigma^*$ is context-free for $j = 1, 2, \dots, q$. Set $L'_j := L_j \cap R$ and observe that each L'_j is context-free. We thus have

$$W' = \bigcap_{j=1}^q L'_j \subseteq R$$

Now consider the Parikh map $\pi: \Sigma^* \rightarrow \mathbb{N}^4$ and observe that $\pi|_R$ is injective. Since W' is poly-context-free, it follows that $\pi(W') \subseteq \mathbb{N}^4$ is semi-linear and there is a regular language $U \subseteq \Sigma^*$ such that

$$(5.2) \quad \pi(U) = \pi(W') = \{(1, n^s, s, s) : s \geq 1\} \subseteq \mathbb{N}^4.$$

One has to be careful since letters in words in U may occur in a very different order than in words of W' . (For example, the words $t^s b t^{-s} b^{-n^s} \in W'$ and $b^{-n^s} (t t^{-1})^s b \in W \setminus W'$ have the same image under the Parikh map.)

Let C be the pumping lemma constant for U . Chose s so that $n^s > n^{s-1} + C$. Then we can find a word $u \in U$ such that $\pi(u) = \pi(t^s b t^{-s} b^{-n^s}) = (1, n^s, s, s) \in \mathbb{N}^4$. Then, by the pumping lemma, u has a decomposition $u = xyz$ where $|xy| \leq C$, $|y| \geq 1$ and (by pumping down) $u' := xz \in U$.

We claim that no word obtained by deleting at least one and no more than C letters from u can have the same Parikh vector as any word in W' . First, we cannot delete b since all words in W' contain exactly one occurrence of b . Suppose we delete $0 < j \leq s$ occurrences of t or t^{-1} . Then we must delete the same number of each to have a word with the same π -image as a word in W' . Let us denote by $0 \leq \ell \leq C$ the number of occurrences of b^{-1} that have also been deleted, so that $\pi(u') = (1, n^s - \ell, s - j, s - j)$. Since $\pi(u') \in \pi(U)$, (5.2) gives us $n^s - \ell = n^{s-j} \leq n^{s-1} < n^s - C$, so that $\ell > C$, a contradiction. Finally, if we only cancel some occurrences of b^{-1} , then (5.2) again yields a contradiction. This shows that W is not poly-context-free. \square

There is a faithful representation of the Baumslag-Solitar group $BS(1, n)$, $n \geq 2$, into $GL(2, \mathbb{Z}[\frac{1}{n}])$ which has been studied in connection with diffeomorphisms of the circle [3, 8]. For each $n \geq 2$ we construct a faithful representation of the group $BS(1, n^2)$ into $SL(2, \mathbb{Z}[\frac{1}{n}])$ by setting

$$(5.3) \quad b \mapsto \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad t \mapsto \begin{pmatrix} n & 0 \\ 0 & \frac{1}{n} \end{pmatrix}.$$

It is easy to verify that the proposed mapping into $SL(2, \mathbb{Z}[\frac{1}{n}])$ preserves the defining relation by multiplying out the matrices for the product $t b t^{-1}$. So the representation (5.3) is well-defined.

Theorem 5.4. *The representation of $BS(1, n^2)$ into $SL(2, \mathbb{Z}[\frac{1}{n}])$ defined in (5.3) is faithful.*

Proof. Let $w \in BS(1, n^2)$ be an arbitrary reduced word. First note that from (5.3) one immediately deduces that

$$b^m \mapsto \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad t^{-\ell} \mapsto \begin{pmatrix} \frac{1}{n^\ell} & 0 \\ 0 & n^\ell \end{pmatrix},$$

for all $m, \ell \in \mathbb{Z}$, so no nonzero power of b or of t goes to the identity. Thus, we can suppose that w is not equal to a power of b or t . Up to replacing w with w^{-1} if necessary, we may suppose that w has nonpositive exponent sum on t . Since the defining relation for $BS(1, n^2)$ can be written $t b = b^{n^2} t$, we can move all occurrences of t^{+1} in w to the right, yielding a word of the form $w_1 t^\alpha$, where w_1 does not contain any positive powers of t and

$\alpha \geq 0$. Thus w is conjugate to $w' := t^\alpha w_1$. Moving t^α to the right, all positive occurrences of t will cancel in w' , since w has nonpositive exponent sum on t . Since w is not a power of b , we have that w or w^{-1} is conjugate to a word of the form $b^{m_1} t^{-\ell_1} b^{m_2} t^{-\ell_2} \dots b^{m_r} t^{-\ell_r}$ where $m_i \in \mathbb{Z}$ and $\ell_i > 0$ for all $i = 1, 2, \dots, r$. The image of such an element is a matrix of the form

$$\begin{pmatrix} \frac{1}{n^\ell} & * \\ 0 & n^\ell \end{pmatrix}$$

where $\ell = \ell_1 + \ell_2 + \dots + \ell_r > 0$, and therefore is not the identity matrix. This shows that (5.3) is faithful. \square

The groups $SL(2, \mathbb{Z}[\frac{1}{n}])$ are interesting here because Serre [18] showed that the groups $SL(2, \mathbb{Z}[\frac{1}{p}])$, p a prime, are free products of two copies of $SL(2, \mathbb{Z})$ amalgamating subgroups of finite index. *However*, the amalgamation is *not* induced by an isomorphism between the two copies and so these groups are *not* doubles. Since a double of $SL(2, \mathbb{Z})$ over a subgroup of finite index is in \mathcal{PG} (cf. Theorem 4.3), the groups $SL(2, \mathbb{Z}[\frac{1}{p}])$ are on the borderline of not being in \mathcal{PG} .

6. SOME DECISION PROBLEMS

Many classical decision problems are already undecidable for direct products of free groups. Mikhailova's theorem [13] shows that $F_2 \times F_2$ has specific finitely generated subgroups with unsolvable *membership problem*. C.F. Miller III [14] showed that $F_3 \times F_3$ contains subgroups with unsolvable *conjugacy problem*. Miller [14] also showed that if $n \geq 5$ then the *generating problem* for $F_n \times F_n$ is undecidable. Thus the conjugacy, membership and generating problems are generally unsolvable for groups in \mathcal{D} .

The *order problem* for a finitely generated group is the problem of deciding whether or not arbitrary elements have infinite order. Holt, Rees, Röver and Thomas [11] prove the interesting result that groups with context-free co-word problem have solvable order problem. We use the main ideas of their proof to establish the same result for groups in \mathcal{BDG} .

Theorem 6.1. *A group in \mathcal{BDG} has solvable order problem.*

Proof. Let $G = \langle X; R \rangle \in \mathcal{BDG}$ and denote by $\Sigma = X \cup X^{-1}$ the associated group alphabet. Since the class of deterministic multipass languages is closed under complementation (cf. Proposition 2.2), the complement $C := \Sigma^* \setminus \text{WP}(G : X; R)$ of the word problem of G is also deterministic multipass. By virtue of Theorem 2.6 we can find context-free languages $L_1, L_2, \dots, L_n \subseteq \Sigma^*$ such that $C = \bigcap_{i=1}^n L_i$. For any $w \in \Sigma^*$ let $w^+ := \{w^n : n \geq 1\} \subseteq \Sigma^*$ denote the regular language consisting of all positive powers of w . Then w represents a group element of infinite order if and only if the multipass language $L_w := (w^+ \cap C)$ equals w^+ . But $L_w = w^+$ if and only if $(w^+ \cap L_i) = w^+$ for all $i = 1, 2, \dots, n$. Since the class of context-free languages is closed under intersection with regular languages, each language $L'_i := w^+ \cap L_i$ is context-free (in fact, a grammar \mathcal{G}'_i for L'_i can be effectively constructed from a grammar \mathcal{G}_i for L_i).

Recall that a language $L \subseteq \Sigma^*$ is said to be *bounded* provided there are words $z_1, z_2, \dots, z_m \in \Sigma^*$ such that $L \subseteq z_1^* z_2^* \dots z_m^*$. Thus w^+ and therefore L'_1, L'_2, \dots, L'_n are bounded context-free languages. Although equality is generally undecidable for context-free languages, results of Ginsburg [5] show that there is an algorithm which, given a bounded context-free language and another context-free language, decides if they are equal. So we can decide if $w^+ = L'_i$ for each $i = 1, 2, \dots, n$ and thus whether or not w represents a group element of infinite order. \square

Note that taking the complement of the word problem is essential for the above proof.

It is well known that the membership problem for deterministic context-free languages is decidable in linear time [10, Section 5.6]. Essentially the same argument gives the following:

Theorem 6.2. *If M is a complete, deterministic multipass automaton, the membership problem for $L(M)$ is solvable in linear time.*

Proof. Since M is complete there is a bound B on the number of ϵ -transitions which M can make before either advancing the tape or erasing the symbol which was on the top of the stack when M began the sequence of ϵ -transitions. Let C be the maximum number of symbols which M can add to the stack on a single transition. Then on a single pass on an input of length n , at most CBn symbols can be added to the stack. So in the complete run of M on the input at most $kCBn$ symbols can be added to the stack and if it takes B moves to erase each symbol then M makes at most kCB^2n transitions. \square

The Cocke-Kasami-Younger algorithm [10, 12] shows that the membership problem for an arbitrary context-free language is decidable in time $\mathcal{O}(n^3)$. Since we need only check subsequent terms in an intersection of context-free languages, it follows that the membership problem for any poly-context-free language is decidable in time $\mathcal{O}(n^3)$.

In formal language theory it is well-known that deciding whether or not the intersection of two deterministic context-free languages is empty is undecidable. That is, there does not exist an algorithm which, when given two deterministic push-down automata M_1 and M_2 over the same input alphabet, decides whether or not $L(M_1) \cap L(M_2) = \emptyset$. The standard construction shows that one can represent valid computations of Turing machines as the intersection of two deterministic context-free languages [12]. Since the intersection of two deterministic context-free languages is a deterministic multipass language, we have:

Proposition 6.3. *The emptiness problem for deterministic multipass languages is undecidable. That is, there does not exist an algorithm which, when given a deterministic multipass automaton M decides whether or not $L(M) = \emptyset$.*

7. THE CLASS \mathcal{D} AND HNN EXTENSIONS

We now show that the class \mathcal{D} of groups that are virtually finitely generated subgroups of direct product of free groups is closed under the class of HNN extensions given in Theorem 4.1. In fact we will prove a slightly more general statement: taking such HNN extensions virtually amounts to taking a direct product with a free group of finite rank.

Theorem 7.1. *Let G be finitely generated group in \mathcal{D} with S a subgroup G of finite index. Let $\varphi: G \rightarrow G$ be an automorphism of finite order of G such that $\varphi(S) = S$. Let H be the HNN-extension*

$$H = \langle G, t; tst^{-1} = \varphi(s), s \in S \rangle.$$

There is a subgroup $E \leq S \leq G$ that is a finite index characteristic subgroup of G . Furthermore for any such subgroup E , H has a finite index subgroup H' isomorphic to $E \times F$, where F is a free group of finite rank.

Proof. Let $E \leq S \leq G \leq H$ be as in the statement of the theorem. The proof is divided into five steps.

Step 1. The existence of E and its normality in H . Suppose that $S \leq G$ has index f . Then taking E to be the intersection of all index f subgroups of G gives the desired property. Since E is characteristic and thus normal in G and φ is an automorphism of G , t normalizes E . E is therefore normalized by a generating set of H and is thus normal in H .

Step 2. The quotient H/E is virtually free. This quotient splits as the HNN extension

$$H/E = \langle \bar{G}, t; t\bar{s}t^{-1} = \bar{\varphi}(\bar{s}), \bar{s} \in \bar{S} \rangle,$$

where \bar{G} denotes the quotient G/E and \bar{s}, \bar{S} denote the image of $s \in S$, the image of the subgroup S in \bar{G} respectively. H/E is therefore the fundamental group of a finite graph of groups with finite vertex groups and therefore (c.f. [18, II.2.6, Proposition 11 and Corollary]) contains a finite index free subgroup F_k of rank k .

Step 3. The structure of the preimage \hat{F}_k as a semidirect product. Denote by \hat{F}_k the preimage of F_k in H via the canonical epimorphism $H \rightarrow H/E$. Then \hat{F}_k has finite index in H and furthermore, since F_k is free, the short exact sequence

$$1 \rightarrow E \rightarrow \hat{F}_k \rightarrow F_k \rightarrow 1$$

splits. Specifically there is a set of elements $\{f_1, \dots, f_k\} \subseteq \hat{F}$ such that

- (1) $\langle f_1, \dots, f_k \rangle \cong F_k$,
- (2) $\hat{F}_k = \langle E, f_1, \dots, f_k \rangle$, and
- (3) $\hat{F}_k \cong E \rtimes_{\rho} \langle f_1, \dots, f_k \rangle$, where $\rho: \langle f_1, \dots, f_k \rangle \rightarrow \text{Aut}(E)$ is the representation induced by conjugation in H . Note that ρ is well-defined since E is normal in \hat{F} .

We will now study the representation ρ more closely.

Step 4. The action of H on E and outer automorphisms. Since E is normal in H there is a natural conjugation representation $\rho_H: H \rightarrow \text{Aut}(E)$. We now study the map $\bar{\rho}_H: H \rightarrow \text{Out}(E)$, where $\text{Out}(E) = \text{Aut}(E)/\text{Inn}(E)$. Our goal is to show that its image is finite.

Denote by ρ_G the restriction of ρ_H to ρ_G . We first note that, since φ is an automorphism of the entire base group G , the following calculation holds for all $k \in \mathbb{Z}$, $g \in G$ and $e \in E$:

$$(7.1) \quad (\varphi^k \circ \rho_G(g))(e) = \varphi^k(geg^{-1}) = \varphi^k(g)\varphi^k(e)\varphi^k(g^{-1}) = (\rho_G(\varphi(g)) \circ \varphi^k)(e).$$

Since every element of H can be written as a product of elements of the form $t^k g$, with $k \in \mathbb{Z}$ and $g \in G$, and since

$$(t^k g) e (g^{-1} t^{-k}) = (\varphi^k \circ \rho_G(g)) (e),$$

the image $\rho_H(H)$ of H in $\text{Aut}(E)$ is generated by elements of the form $\varphi^k \circ \rho_G(g)$. The identity (7.1) enables us to collect φ^k 's to the left, so we can express every element of $\rho_H(H)$ in a normal form:

$$\varphi^n \circ \rho_G(g),$$

with $1 \leq n \leq \text{order}(\varphi)$ and $g \in G$. If $g' = ge$, for some $g \in G, e \in E$ then $\rho_G(g') \text{Inn}(E) = \rho_G(g) \text{Inn}(E)$. It follows that if $g_1, \dots, g_{[G:E]}$ is a finite set of left coset representatives of E in G , then the finite set of automorphisms

$$\{\varphi^i \circ \rho_G(g_j) \mid 1 \leq i \leq \text{order}(\varphi), 1 \leq j \leq [G : E]\}$$

gives a complete set of representatives of the image of H in $\text{Out}(E)$. The image of $\bar{\rho}_H : H \rightarrow \text{Out}(E)$ is therefore finite.

Step 5. Untwisting the semidirect product. The twisting representation ρ in $\hat{F}_k \cong E \rtimes_{\rho} \langle f_1, \dots, f_k \rangle$ is the restriction of ρ_H to the subgroup $\langle f_1, \dots, f_k \rangle$. It follows from the previous part that the image of $\bar{\rho} : \langle f_1, \dots, f_k \rangle \rightarrow \text{Out}(E)$ is finite; thus the kernel $K = \ker(\bar{\rho})$ is of finite index in $\langle f_1, \dots, f_k \rangle$. Let $X = \{k_1, \dots, k_m\}$ be a free generating set for K . On one hand the subgroup $H' = \langle E, k_1, \dots, k_m \rangle \cong E \rtimes_{\rho|_K} K$ is of finite index in \hat{F}_k , and therefore in H . On the other hand, since $K = \ker(\bar{\rho})$, for each $k_i \in X$ there exists some $e_i \in E$ such that for all $e \in E$,

$$k_i e k_i^{-1} = e_i e e_i^{-1}.$$

It follows that

$$H' = \langle E, k_1 e_1^{-1}, \dots, k_m e_m^{-1} \rangle \cong E \times F,$$

where $F \cong K$ is a free group of rank m . □

For the proof to work, it is essential that the isomorphism $\varphi : E \rightarrow E$ in the HNN extension extends to an isomorphism $\varphi : G \rightarrow G$. If this is not the case, then the identity (7.1) need not hold, which in turn prevents rewriting into normal form. In particular the image $\bar{\rho}_H(H)$ could be infinite, which makes it impossible to virtually untwist the semidirect product in Step 5.

We point out that the class \mathcal{D} is closed under taking quotients by finite normal subgroups. If G has a subgroup S of finite index which is a subgroup of a direct product of free groups then S is torsion-free. So $N \cap S = \{1\}$ and S is embedded in the quotient G/S and again has finite index.

In conclusion, if we start with a virtually free group, none of the known closure properties of the class \mathcal{PG} which we have discussed take us outside the class \mathcal{D} . This seems fairly strong evidence in favor of Brough's Conjecture.

Acknowledgments. We wish to thank Ralph Strebel for very helpful discussions and his interest in our work as well as for some simplifications of an earlier version of the proof of

Theorem 7.1. We also express our gratitude to the two referees for their careful reading of our manuscript and their valuable comments.

REFERENCES

- [1] A.V. Anisimov, *Über Gruppen-Sprachen*, Kibernetika **4** (1971), 18–24.
- [2] T. Brough, *Groups with poly-context-free word problem*, arXiv: 1104.1806, 10 April, 2011.
- [3] L. Burslem and A. Wilkinson, *Global rigidity of solvable group actions on S^1* , Geom. Topology **8** (2004), 877–924.
- [4] T. Ceccherini-Silberstein, M. Coornaert, F. Fiorenzi and P.E. Schupp, *Groups, Graphs, Languages, Automata, Games and Second-order Monadic Logic*, European J. Combinatorics, **33** (2012), 1330–1368.
- [5] S. Ginsburg, *The Mathematical Theory of Context-free Languages*, McGraw-Hill, New York, 1966.
- [6] S. Ginsburg and E.H. Spanier, *Semigroups, Presburger formulas and languages*, Pacific J. Mathematics **16** (1966), 285–296.
- [7] I. Gorun, *On intersections of context-free languages*, Fund. Inform. **3** (1980), no. 4, 491–496.
- [8] N. Guelman and I. Liousse, *C^1 actions of Baumslag-Solitar groups on S^1* , Algebr. Geom. Topol. **11** (2011), 1701–1707.
- [9] R.H. Haring-Smith, *Groups and simple languages*, Trans. Am. Math. Soc. **279** (1983), 337–356.
- [10] M.A. Harrison, *Introduction to Formal Language Theory*, Addison-Wesley, Reading, MA, 1978.
- [11] D.F. Holt, S. Rees, C. Rover and R.M. Thomas, *Groups with context-free co-word problem* J. London Math. Soc. (2) **71** (2005), 643–657.
- [12] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading, MA, 1979.
- [13] R.C. Lyndon and P.E. Schupp, *Combinatorial Group Theory*, Springer-Verlag Classics in Mathematics, New York, 2000.
- [14] C.F. Miller III, *On Group-theoretic Decision Problems and Their Classification*, Annals of Mathematics Studies **68**, Princeton University Press, Princeton, 1971.
- [15] D.E. Muller and P.E. Schupp, *Groups, the theory of ends and context-free languages*, J. Comput. System Sci. **26** (1983), 295–310.
- [16] R.J. Parikh, *Language-generating devices*, MIT Res. Lab. Electron. Quart. Prog. Report, **60** (1961), 199–212.
- [17] R.J. Parikh, *On context-free languages*, J. Assoc. Computing Machinery **13** (1966), 570–581.
- [18] J.P. Serre, *Trees*, Springer Monographs in Mathematics, Springer-Verlag, Berlin, 2003, Translated from the French original by John Stillwell, Corrected 2nd printing of the 1980 English translation.
- [19] J. Stallings, *A finitely presented group whose 3-dimensional homology is not finitely generated*, Am. J. Math. **85** (1963), 541–543.
- [20] D. Wotschke, *The Boolean closures of the deterministic and nondeterministic context-free languages*, in Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, New York, (1973), 113–121.
- [21] D. Wotschke, *A characterization of Boolean closures of families of languages*, in Lecture Notes in Computer Science, Springer-Verlag, New York, (1973), 191–200.

DIPARTIMENTO DI INGEGNERIA, UNIVERSITÀ DEL SANNIO, C.SO GARIBALDI 107, 82100 BENEVENTO, ITALY

E-mail address: `tceccher@mat.uniroma3.it`

INSTITUT DE RECHERCHE MATHÉMATIQUE AVANCÉE, UMR 7501, UNIVERSITÉ DE STRASBOURG ET CNRS, 7 RUE RENÉ-DESCARTES, 67000 STRASBOURG, FRANCE

E-mail address: `coornaert@math.unistra.fr`

LABORATOIRE DE RECHERCHE EN INFORMATIQUE, UNIVERSITÉ PARIS-SUD 11, 91405 ORSAY CEDEX, FRANCE

E-mail address: `fiorenzi@lri.fr`

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN, 1409 W. GREEN STREET (MC-382) URBANA, ILLINOIS 61801- 2975, USA

E-mail address: `schupp@math.uiuc.edu`

DEPARTMENT OF MATHEMATICAL SCIENCES, STEVENS INSTITUTE OF TECHNOLOGY, 1 CASTLE POINT TERRACE HOBOKEN, NEW JERSEY 07030, USA

E-mail address: `nicholas.touikan@gmail.com`